Identication of fuzzy function via interval analysis

Vincent Vigneron, Madalina Olteanu, Hichem Maaref

Identification of fuzzy function via interval analysis

V. Vigneron [a,*] M. Olteanu [b] H. Maaref [a]

[a] *IBISC, CNRS FRE 2873, 40 rue du Pelvoux, CE 1455, 91020 Evry-Courcouronnes cedex, France.*
Email: vincent.vigneron,hichem.maaref@ibisc.univ-evry.fr

[b] *MATISSE-SAMOS CNRS UMR 8595, 90 rue de Tolbiac, 75634 Paris cedex 13, France.*
Email: vigneron,kallel@univ-paris1.fr

Contents

1	Introduction	2
2	A refresher on parameter estimation	3
3	Identification method	4
3.1	Core of the modeling	4
3.2	Set characterization for each α-cut	6
3.3	Discussion	8
4	Fuzzy systems as universal approximators	10
5	Numerical experiments	11
5.1	Discussion	15
6	Conclusion	18
References		19

Abstract

A number of techniques have been introduced to construct fuzzy models from measured data. One of the most common is the use of mathematical parametric models. In this paper, a new approach based on interval analysis is proposed to compute guaranteed estimates of suitable characteristics of the set of all values of the fuzzy parameter vector such that the error between experimental data and the model outputs belongs to some predefined feasible set. Subpavings consisting of boxes with nonzero width are used to encompass all the acceptable values of the parameter vector. The problem of estimating the parameters of the model is viewed as one of set inversion, which is solved in an approximate but guaranteed way with the tools of interval analysis. The estimation task is formulated here as a constrained optimization problem. Our approach emphasizes the use of interval mathematics for fuzzy representation, which is especially suited to nonlinear models, a situation where most available methods fail to provide any guarantee on the results. An algorithm is proposed, which makes it possible to obtain all fuzzy parameter vectors that are consistent with the data. Properties of this algorithm are established and illustrated on a simple example.

Key words: Fuzzy function approximation, fuzzy interval, identification, estimation, constraint satisfaction problem, interval analysis.

5th October 2006

Notations

Throughout the text, the following conventions are used. Lower case letters in italics such as x or y_i denote scalar variables and elements of vectors. Vectors are printed in bold. A row vector is denoted by the transpose operator, *i.e.* $\mathbf{x}^T$. Uppercase bold characters denote matrices, for instance $\mathbf{X}$. Uppercase italic letters such as A denote crisp and fuzzy sets. The term "crisp" is used as opposite to fuzzy. A linguistic variable (a variable whose values are fuzzy sets) is denoted by $\tilde{x}$.

1 Introduction

Consider the situation where we have a model, which acts as a function f mapping inputs X to outputs Y. This model f might be quite complex, with multiple input parameters and with different kinds of uncertainty represented on them: information available on inputs may be rich or sparse, so-called "aleatory" and may be made known through objective measurements. Mathematically, inputs might be represented as probability or possibility distributions, by a strong or sparse statistical collection of data points, by simple intervals, or even by non-quantified linguistic expressions.
Recent works has focused on the idea of constructing fuzzy systems via a finite set of input-output training data in order to perform function approximation [1,2]. This focus is particularly important since many problems in estimation and identification can be formulated as function approximation problems.
Several results [3,4] show that *interval analysis* methods can be directly adapted to fuzzy interval computation where end point of intervals are given by the increasing and decreasing parts of the membership function (MF) of the fuzzy interval. Calculations with fuzzy intervals extend interval analysis. Generally, it consists in applying interval analysis to all *cuts* of the fuzzy intervals. One drawback of this process lie in the fact that the interval algorithm has to be completely executed for each α-cut [5].
So, given f, how can we propagate the uncertainty on the input space through f to the output space? Moreover, how can we do so in a way which respects all the original uncertainty quantifications as provided, making no unnecessary assumptions? How can we do such in a way which uses *only*, but *all of* what we are given?
In this paper, we propose an approach for computing functions of fuzzy intervals, assuming that experimental points (both inputs and outputs) are modeled as fuzzy numbers. Our method makes exact computations, resorting to α-cuts.
The contribution of this article is three-fold: first, it is shown that interval analysis can be directly applied to perform fuzzy interval analysis, yielding closed form expressions of the results. Second, a C++ library is presented which implements the presented techniques. The program and the library can be obtained from the authors on request. Third, an illustrative example of a SISO (Single-Input, Single-Output) nonlinear dynamic process is given. It is seen that the fuzzy identification approach leads to a significative improvement in comparison with standard linear identification.
The paper is organized as follows. Section 2 describes the solution of the problem, *i.e.* how interval analysis can be used in the context of fuzzy identification. Related works are discussed in section 4. The practical significance of the computational

* Corresponding author.

results is pointed out through section 3.1 and an algorithm is proposed. Section 5 illustrates it on a case study with the help of some numerical examples.

2 A refresher on parameter estimation

Consider some function $f : \mathcal{X} \subset \mathbb{R}^n \to \mathcal{Y} \subset \mathbb{R}^s$, where $\mathcal{Y}$ is a bounded set and suppose we wish to construct a *fuzzy* system $g : \mathbb{X} \subset \mathcal{X} \to \mathbb{Y} \subset \mathcal{Y}$, where $\mathbb{X}$ and $\mathbb{Y}$ are some domains of interest, by choosing a parameter vector $\mathbf{p} \in \mathbb{R}^p$ so that, mathematically speaking

$$\mathbf{y} = f(\mathbf{x}) = g(\mathbf{x}, \mathbf{p}) + e(\mathbf{p}), \tag{1}$$

for all $\mathbf{x} \in \mathbb{X}$ and $\mathbf{y} \in \mathbb{Y}$, where the error in approximation $e(\mathbf{p})$ is as small as possible.

We suppose that all that is available to choose the parameters $\mathbf{p}$ in g is some part of the *unknown* function f in the form of the input-output data pair associations. The ith input-output pair for the system f is denoted by $(\mathbf{x}_i, \mathbf{y}_i)$, where $\mathbf{x}_i \in \mathbb{X}, \mathbf{y}_i \in \mathbb{Y}$ and $\mathbf{y}_i = f(\mathbf{x}_i)$. This may correspond for instance to $n + s$ scalar measurements corresponding to various experimental conditions on a static process or on a dynamical one. The row vector $\mathbf{z}_i = (\mathbf{x}_i^T \mathbf{y}_i^T) \in \mathbb{R}^{n+s}$ denotes one particular data sample. Stacking N consecutive samples on top of each other gives the data matrix

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_N \end{bmatrix} \in \mathbb{X} \times \mathbb{Y}. \tag{2}$$

The purpose of parameter estimation is to find $\mathbf{p}$ such that $g(\mathbf{x}, \mathbf{p})$ fits $\mathbf{y}$ best in a sense to be specified, for instance. In [6], the parameters are considered admissible if the error $e(\mathbf{p})$ belongs to some prior compact set of admissible error $\mathbb{E} \subset \mathbb{R}^s$. $\mathbb{E}$ may be for instance the box defined as

$$\mathbb{E} = \{e | e^- \leq e \leq e^+\}, \tag{3}$$

where e^+ and e^- are some prior bounds. Other type of compact sets could be considered as more general polytopes. One is then interest in finding the set $\mathbb{S}$ of all values of $\mathbf{p}$ such that the error is admissible, *i.e.* $\mathbb{S} = \{\mathbf{p} | e(\mathbf{p}) \in \mathbb{E}\}$. This set has been called *membership set*, *likelihood set* and *posterior feasible set*. If the data have been generated by a fuzzy model $g(\mathbf{x}, \mathbf{p}^*)$, where $\mathbf{p}^*$ is some true value of the parameters and if $e(\mathbf{p}^*) \in \mathbb{E}$, then $\mathbb{S}$ contains $\mathbf{p}^*$. Thus, $\mathbb{S}$ provides us an accurate description of the uncertainty with which $\mathbf{p}^*$ is estimated [6].

If the reciprocal of g exists and is denoted g^{-1}, $\mathbb{S}$ is defined as $\mathbb{S} = g^{-1}(\mathbf{y} - \mathbb{E}) = g^{-1}(\mathbb{Y})$, where $\mathbb{Y} = \mathbf{y} - \mathbb{E}$ is the *measurement set*. In other words, for any $\mathbf{p} \in \mathbb{S}$, there exists $e \in \mathbb{E}$ such that $\mathbf{y} = g(\mathbf{x}, \mathbf{p}) + e$.

System identification (*i.e.* function approximation) amounts to adjusting $\mathbf{p}$ using information from $\mathbf{Z}$ so that $g(\mathbf{x}, \mathbf{p}) \approx f(\mathbf{x}), \forall \mathbf{x} \in \mathbb{X}$. Measured and model outputs never match perfectly in practice, but differ as $e(\mathbf{p})$. An obvious modeling goal must be that this discrepancy is "small" in some sense that is archieved by the value of the *approximation error* we wish to bound. Such a bound is, for example

$$\sup_{\mathbf{x} \in \mathbb{X}} \{f(\mathbf{x}) - g(\mathbf{x}, \mathbf{p})\}, \tag{4}$$

which requires that f be known everywhere. The problem is that we only know the part of f given by $\mathbf{Z}$, and it is the only evaluation we can make based on known information.
When g is not affine in $\mathbf{p}$, one may linearizing it around some value of the parameters estimated beforehand and then using any method for linear models. Scanning the parametric space using random search offers no guarantee as to the global nature of the results obtained [7].
A thorough examination of the error $e(\mathbf{p})$ that are commonly used reveals that both *imprecision* and *uncertainty* play a key role in the parameter value formulation.
Our approach departs from a different position by considering fuzziness in the experimental points while using a crisp model: indeed, it is reasonnable to assume that a large part of estimation deviations of $\mathbf{p}$ may come from imperfection in input assesment, together with the fuzziness of the parameters of the model [8].

3 Identification method

3.1 Core of the modeling

The structure of the model, *i.e.* the form of g and the matrix $\mathbf{Z}$, are determined by the user on the basis of prior knowledge and/or by comparing several candidate structures in terms of the prediction error. Once the structure is fixed, the parameters of the model can be estimated. Modeling possible values of variables by means of real intervals (as in interval computation) accounts for some uncertainty, but we can be more precise by modeling uncertainty on a variable x_i by means of a fuzzy interval X_i. Then, one way to compute the possible fuzzy range P of $\mathbf{p}$ is to decompose the problem in terms of $\alpha-$cuts, and then to apply an interval analysis method. The main drawback of this approach is that we compute only an approximation of P, and that, for each $\alpha-$cut, the interval algorithm has to be completely executed [9].
The $\alpha-$cut of A denoted by A^α, for any scalar $\alpha \in [0,1]$, is a *crisp set* $\{X \mid \mu_A(X) \geq \alpha\}$, where $\mu_A(X)$ is the MF for the fuzzy set A, which can formally be written as $\mu_A^\alpha = \alpha\mu_A(X),\ X \in \mathbb{X}$. μ_A^α is defined as follow:

$$\mu_A^\alpha(X) = \begin{cases} \alpha, & \forall X \in A^\alpha \\ 0, & \text{otherwise,} \end{cases} \qquad \forall \alpha \in [0,1]. \tag{5}$$

Note that the $\alpha-$cut of a fuzzy interval is a "classical" interval. This decomposition by α-cut of A can be used to compute the function on fuzzy intervals: for a $n-$tuple of variables $\mathbf{x} = \{x_1, \ldots, x_n\}$, we have

$$[g(x_1, \ldots, x_n; \mathbf{p})]_\alpha = g([x_1]_\alpha, \ldots, [x_n]_\alpha; [\mathbf{p}]_\alpha). \tag{6}$$

A box or *vector* of intervals $[\mathbf{x}]$ is the cartesian product of n intervals and is noted $[x_1] \times \ldots \times [x_n]$, with $[x_i] = [\underline{x_i}, \overline{x_i}], i = 1, \ldots, n$. The ith interval component $[x_i]$ is the projection of $[\mathbf{x}]$ onto the ith axis.
Consider a fuzzily described system with fuzzy input and output X and Y resp., which are readings from unreliable sensor ("noisy data"). Let $\{x_1, \ldots, x_n\}$ be a tuple of $n-$independent variables restricted to their fuzzy intervals $X_1, \ldots, X_n$, defined by their MF $\mu_{X_1}, \ldots, \mu_{X_n}$, and Y the fuzzy set of the possible values of the variable

$y = g(\mathbf{x};\mathbf{p})$. To simplify exposition, we shall only consider output errors :

$$e_i(\mathbf{p}) = y_i - g(\mathbf{x}_i;\mathbf{p}),\ i = 1,\ldots,s \tag{7}$$

but other types of errors could be considered as well. We assume that these errors should satisfy $\underline{e_i} \le e_i(\mathbf{p}) \le \overline{e_i},\ i = 1,\ldots,s$ to be acceptable, where $\underline{e_i}$ and $\overline{e_i}$ are known lower and upper prior bounds of the approximation error, that results from technical specifications or indicate how far we can go in accepting discrepancies between our data and model outputs. Note that $e_i(\mathbf{p})$ is a fuzzy real number bounded by two profiles obtained by the increasing and decreasing parts of the error MF.
Let $\mathbf{y}$ be the vector of all data $y_i, i = 1,\ldots,s$ collected on a given system, and $\mathbf{g}(\mathbf{x};\mathbf{p})$ be the vector of all corresponding model outputs $g(\mathbf{x}_i;\mathbf{p}),\ i = 1,\ldots,s$. The vector of all output errors between the actual output $g(\mathbf{x};\mathbf{p})$ and the target $\mathbf{y}$ can then be written as

$$\mathbf{e}(\mathbf{p}) = \mathbf{y} - g(\mathbf{x};\mathbf{p}), \tag{8}$$

where $\mathbf{p},\mathbf{y}$ and $\mathbf{x}$ are vectors of fuzzy real numbers. Picked in each $\alpha-$cut, equation (8) gives:

$$[\mathbf{e}(\mathbf{p})]_\alpha = [\mathbf{y}]_\alpha - g([\mathbf{x}]_\alpha;[\mathbf{p}]_\alpha), \tag{9}$$

The model (9) is acceptable if $\mathbf{p}$ is such that $\mathbf{e} \in \mathbb{E}$, *i.e.* the set of all error vectors $\mathbf{e}$ satisfies $\underline{\mathbf{e}} \le \mathbf{e} \le \overline{\mathbf{e}}$ where $\underline{\mathbf{e}}$ and $\overline{\mathbf{e}}$ are known. Estimating fuzzy interval $[\mathbf{p}]_\alpha$ amounts to look at the level α for the set of all admissible values of P^α that are consistent with (9), *i.e.* errors should satisfy

$$[\mathbf{y}]_\alpha - [g]([\mathbf{x}]_\alpha;[\mathbf{p}]_\alpha) \in [\mathbf{e}]([\mathbf{p}]_\alpha),\ \alpha \in [0,1], \tag{10}$$

Equation (10) relies on the assumption that there exists an *inclusion function* $[g]$ of g that returns an *enveloping box* guaranteed to contain the image by g of any given box $[\mathbf{x}]$ included in the domain of g. $[\mathbf{g}]([\mathbf{x}])$ is a box such that

$$\mathbf{g}([\mathbf{x}]) \subset [\mathbf{g}]([\mathbf{x}]), \tag{11}$$

It is easy to compute for usual elementary functions that can be obtained by composition of elementary operations such as $+,-,\times,/,\exp,\tan,\sin,\ldots$ by replacing each of these elementary operations by their inclusion function in the formal expression of g.
When no efficient algorithm exists for the computation of $[g]$, it can be approximated by an inclusion function $\mathbb{G} : \mathbb{R}^n \to \mathbb{R}^p$ satisfying equation (11), such that:

$$w([\mathbf{x}]) \to 0 \Rightarrow w([\mathbb{G}]([\mathbf{x}])) \to 0. \tag{12}$$

where the width $w([\mathbf{x}])$ of the box $[\mathbf{x}]$ is the length of its largest side(s). It must be noted that the algorithm presented in this paper is guaranteed to converge only if (12) is valid.

Figure 1 illustrates conditions (11) and (12).
The *feasible* fuzzy set for $\mathbf{p}$ is therefore simply given by

$$\mathbb{S} = \bigcup_{\alpha\in[0,1]} \mathbb{S}_\alpha = \bigcup_{\alpha\in[0,1]} \mathbf{g}^{-1}([\mathbf{y} - \mathbb{E}]_\alpha). \tag{13}$$

with evident notations. Let $\widehat{\mathbb{S}} = \{\mathbf{p} | g([\mathbf{x}];[\mathbf{p}]) \in \mathbb{Y}\}$. Then

$$g([\mathbf{x}];[\mathbf{p}]) \in \mathbb{Y} \leftrightarrow \mathbf{p} \in g^{-1}(\mathbb{Y}) \quad \text{and} \quad \mathbf{p} \in [\mathbf{p}](0) \tag{14}$$

$$\leftrightarrow \quad \mathbf{p} \in [\mathbf{p}](0) \cap g^{-1}(\mathbb{Y}) \tag{15}$$

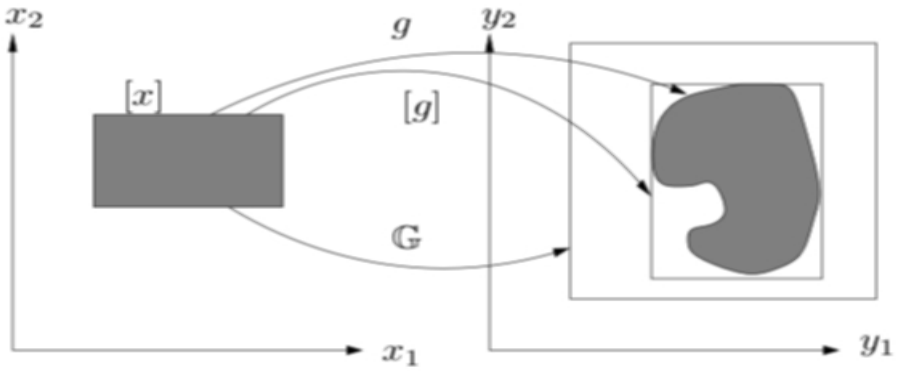


Figure 1. Minimal inclusion function $[g]$ and inclusion function $\mathbb{G}$ of a function g.

where $[\mathbf{p}](0)$ is the search domain. Thus

$$\widehat{\mathbb{S}} = [\mathbf{p}](0) \cap g^{-1}(\mathbb{Y}), \tag{16}$$

and characterizing $\widehat{\mathbb{S}}$ is a *set inversion problem*.
From (13), we shall say that $[\mathbf{p}]$ is *feasible* if $[\mathbf{p}] \subset \mathbb{S}$, *unfeasible* if $[\mathbf{p}] \cap \mathbb{S} = \varnothing$, else $[\mathbf{p}]$ is *ambiguous*.
To perform fuzzy approximation in an approximate but guaranteed way, the problem is decomposed in terms of α-cuts and an interval analysis method is applied to compute the possible fuzzy range $[\mathbf{p}]_\alpha$ of $\mathbf{p}$ at each level α. The interval computation stage is clarified in the following.

3.2 Set characterization for each α-cut

Methods making it possible to implement interval analysis are relatively few and date from the Nineties, among which, one can quote the algorithm of Moore [10] and SIVIA proposed by Jaulin [11]. Most methods for estimating parameters are based on computations performed at point values of the parameter vector. The main interest in the notion of paving is to make it possible to replace point values by subsets of the parameter space. For simplicity, we will use pavings based upon boxes. By convention, a *paving* of a compact subset $\{\mathbb{P}\} \subset \mathbb{R}^n$ is a set of non overlapping boxes with nonzero width such that the union of these boxes corresponds to $\{\mathbb{P}\}$. A subpaving $\mathbb{K}$ of $\mathbb{P}$ is a subset of $\mathbb{P}$. The characteristic of $\mathbb{S}_\alpha$ defined in (13) that we shall consider more specifically is its enveloping box $[\mathbb{S}_\alpha]$ so that the MF of $\mathbb{S}$ can be exactly recoved from the extremes of the $\mathbb{S}_\alpha$. Upon completion, the algorithm encloses $\mathbb{S}_\alpha$ between two compact sets corresponding to 2 subpavings. Our algorithm can be applied to any model structure for which an inclusion function can be computed. Let $\mathbb{E}$ be the feasible error set. Initialisation is performed by setting $\mathbb{Y}_\alpha = \mathbf{y}_\alpha - \mathbb{E}$. The principle is as follows:

a). Define an initial box of interest $[\mathbf{p}]_\alpha(0)$ within which the search will be performed
b). Compute a paving $\{\mathbb{P}\}$ of $[\mathbf{p}]_\alpha(0)$
c). Compute the image $g([\mathbf{x}];[\mathbf{p}]_\alpha)$ for each box of this paving. Three situations must then be considered (see fig. 2).

$[g]([\mathbf{x}]_\alpha;[\mathbf{p}]_\alpha) \subset [\mathbb{Y}]_\alpha \Rightarrow [\mathbf{p}]_\alpha \subset \mathbb{S}_\alpha$ so that $[\mathbb{Y}]_\alpha$ is *feasible*

$[g]([\mathbf{x}]_\alpha;[\mathbf{p}]_\alpha) \cap [\mathbb{Y}]_\alpha = \varnothing \Rightarrow [\mathbf{p}]_\alpha \cap \mathbb{S}_\alpha = \varnothing$ so that $[\mathbf{p}]_\alpha$ is *unfeasible.*

otherwise, $[p]_\alpha$ is *indetermined.*

The exploration algorithm performs a recursive implementation of the principle that has just been described: a *bisection algorithm* splits each box of the subpaving into smaller boxes whenever needed until the width of the box becomes smaller than some tolerance parameter ϵ to be specified by the user. Cutting is carried out again as long as the boxes contain solutions or stops if the boxes do not contain any.

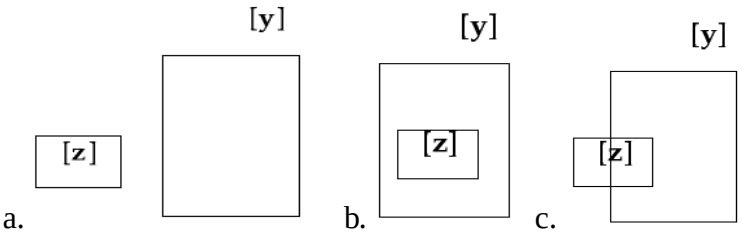


Figure 2. Feasibility of boxes: a test function makes it possible to distinguish the cases (a-c) represented in this figure. Let [**z**] and [**y**] be two boxes. Suppose that [**n**] = [**z**]∩[**y**]. a) [**n**] = ∅ means that [**z**] has an empty intersection with [**y**]. b) [**n**] = [**z**], so that [**z**] is included inside [**y**]. c) [**n**] ⊂ [**z**] and [**n**] ≠ [**z**]. [**z**] intersects [**y**], so that we cannot conclude. The box [**z**] can be split again.

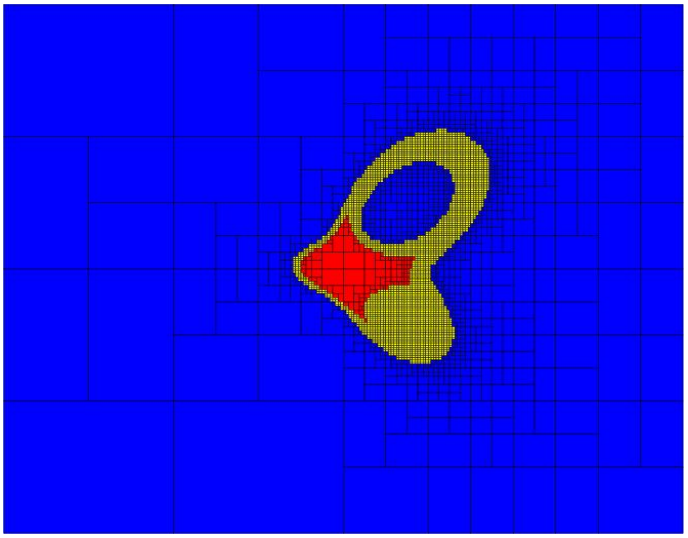
We shall partition $\mathbb{P}$ iteratively into three subpavings $\underline{\mathbb{S}}_\alpha, \mathbb{S}_{\text{out}\alpha}$ and $\overline{\mathbb{S}}_\alpha$ corresponding respectively to the sets of all feasible, unfeasible and indetermined boxes (see fig. 3). These subpavings satisfy the following relations :

1. $\{\underline{\mathbb{S}}_\alpha\} \subset \mathbb{S}_\alpha \subset \{\underline{\mathbb{S}}_\alpha\} \cup \{\overline{\mathbb{S}}_\alpha\}$
2. $\text{vol}(\{\underline{\mathbb{S}}_\alpha\}) \leq \text{vol}(\mathbb{S}_\alpha) \leq \text{vol}(\{\underline{\mathbb{S}}_\alpha\}) + \text{vol}(\{\overline{\mathbb{S}}_\alpha\})$
3. $[\{\underline{\mathbb{S}}_\alpha\}] \subset [\mathbb{S}_\alpha] \subset [\{\underline{\mathbb{S}}_\alpha\}] \cup [\{\overline{\mathbb{S}}_\alpha\}]$

Provided that $\mathbb{S}_\alpha$ is full, this means that the pair $\{\underline{\mathbb{S}}_\alpha;\overline{\mathbb{S}}_\alpha\}$ defines a neighbourhood $\partial\mathbb{S}_\alpha \triangleq \overline{\mathbb{S}}_\alpha \backslash \underline{\mathbb{S}}_\alpha$ of $\mathbb{S}_\alpha$ with a diameter that can be chosen arbitrarily small.
The previous algorithm makes an extensive use of a *stack* L of boxes, *i.e.* a dynamical structure on which only 3 operations are possible: at any time, one may put an element on top of the list, removed the top element or test the stack for emptiness. We define the required accuracy ϵ for the paving $\mathbb{P}$ as the maximum widh that an indetermined box can have. In the following, the principal plane of a box is the symmetry plane of this box that is orthogonal to the axis $i \in \{j | w([\mathbf{p}]_\alpha) = w([p_j]_\alpha)\}$, where the operator "width" $w([\cdot])$ of a box is the length of its largest side.
Let $[\mathbf{p}]_\alpha(0)$ be the box considered at iteration k. Initialisation is performed by setting $k = 0, \mathsf{L} = \varnothing, \underline{\mathbb{S}} = \overline{\mathbb{S}} = \varnothing$. The recursive algorithm can be described as follows:

The union of all the boxes in the list L returned by the program contains $\mathbb{S}$; the partition $\mathbb{P}$ consisting in feasible, unfeasible and indetermined boxes can be plotted in the parameter space in the case the space dimension is less than 4 (see fig. 3).

a.



b.

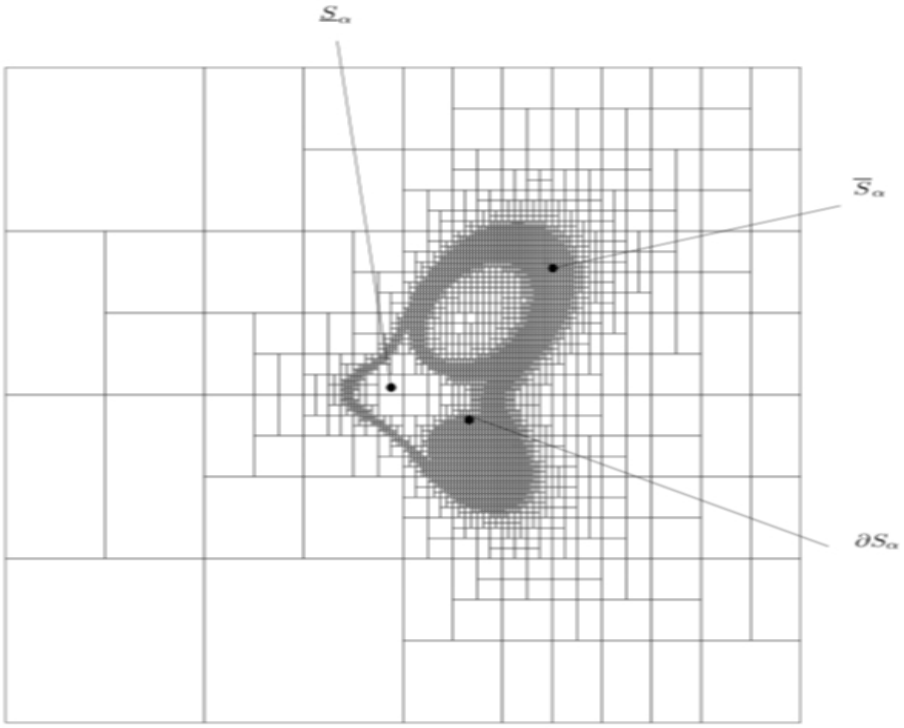


Figure 3. a. Regular paving of a box: accepted, rejected and indetermined subpavings are respectively coloured in red, blue and yellow. b. $\{\underline{\mathbb{S}}_\alpha$ and $\overline{\mathbb{S}}_\alpha\}$ brackets the portion of $\mathbb{S}$ contained in [**p**](**0**).

3.3 Discussion

The method of *set characterization* introduced in section 3.2 appeal some comments.
Upon completion, this approach encompasses *all* the acceptable values of the parameter vector in a set that is fully characterized when the solver BISECT extends to each α-cut, for $\alpha \in [0,1]$: $\underline{\mathbb{S}}$ and $\overline{\mathbb{S}}$ will tend to $\mathbb{S}$ from within and without when

Algorithm BISECT
INPUTS
data: $\mathbf{y}$
inclusion function: $[g]([\cdot])$
feasible error set: $\mathbb{E}$
prior feasible box: $[\mathbf{p}]_\alpha(0)$
accuracy for the paving: ϵ
INITIALIZATION
$\mathbb{Y} = \mathbf{y} - \mathbb{E}$;
stack: $L = \varnothing$
iteration: $k = 0$
$[\mathbf{p}] = [\mathbf{p}](0)$;
ITERATION k
step 1: if $[g]([\mathbf{x}]_\alpha; [\mathbf{p}]_\alpha(k)) \subset [\mathbb{Y}]_\alpha$ then $\underline{\mathbb{S}}_\alpha := \underline{\mathbb{S}}_\alpha \cup [\mathbf{p}]_\alpha$ and $\overline{\mathbb{S}}_\alpha := \overline{\mathbb{S}}_\alpha \cup [\mathbf{p}]$;
step 2: else if $[g]([\mathbf{x}]_\alpha; [\mathbf{p}]_\alpha(k)) \cap [\mathbb{Y}]_\alpha = \varnothing$, then $\underline{\mathbb{S}}_{\text{out}} := \underline{\mathbb{S}}_{\text{out}} \cup [\mathbf{p}]_\alpha$; then unstack $[\mathbf{p}]_\alpha(k)$ as unfeasible;
step 3: else if $w([\mathbf{p}]_\alpha(k)) \leq \epsilon$, then $\overline{\mathbb{S}}_\alpha := \overline{\mathbb{S}}_\alpha \cup [\mathbf{p}]_\alpha(k)$;
else cut $[\mathbf{p}]_\alpha(k)$ along the principal plane and stack the resulting boxes in L.
step 4: if the stack is not empty, then unstack and store the resulting box in $[\mathbf{p}]_\alpha(k+1)$;
$k = k+1$; go to step 1;
End

Table 1
Recursive implementation of the bisection algorithm.

$\epsilon \to 0$. Since $\overline{\mathbb{S}}$ is a finite union of boxes guaranteed to contain the portion of $\mathbb{S}$ of interest, it is very convenient for implementing set-theoretic manipulations [12,6]. The advantages of this approach are threefold:

(i) no assumption is made on the image fonction g,
(ii) no statistical assumption on the error modeling is required,
(iii) any bounded error can be treated independently from its origin (modeling and/or measurement error).

An other advantage of the proposed approach is that the input-output roles of the fuzzy sets X and Y can be reverse since the linking function $\mathbf{g} : X \to Y$ can be run *forwardly* as well as *backwardly* thanks to interval analysis. Subpavings form a useful class of objects for manipulating fuzzy approximations.
The algorithm requires a possibly very large search box $[\mathbf{p}](0)$ to which $\mathbb{S}$ is guaranteed to belong. Solvers partition the search box into an union of boxes (the *paving*), with guaranteed error bounds (*i.e.* mathematically valid) [7] (see section 3.2). The paving is generally built by the solver itself. The results returned by the solver depend only on the results obtained for each box of the paving. A computer program can represent a set of (eventually disjoint) intervals as a *list* L. The precision of the solver is controlled by coefficients specifying, for example, the width ϵ of the smallest boxes of the paving, or the accuracy in the localization of a global optimum. The computing time of the solver can increase quickly with the dimension and size of the list L.

Special care must be taken to avoid memorizing unnecessary information, otherwise the quantity of memory required to store the paving of $\mathbb{S}$ will increase linearily at each iteration, which may result into a memory overflow even for problems of modest dimension.
One can observed that the parameter space is not isotropic because the sensitivities of g with respect to the various components of $\mathbf{p}$ are not of the same order of magnitude. The basic bisection technique suggested in Tab. 1 may not be efficient enough. The problem is then to choose the fastest bisection policy that results in a convergence as rapid as possible. Jaulin *et al.* [6] suggests the bisections of $[\mathbf{p}]$ into boxes $[\mathbf{p_1}]$ and $[\mathbf{p_2}]$ that minimizes $\text{vol}(g([\mathbf{x}];[\mathbf{p_1}]))+\text{vol}(g([\mathbf{x}];[\mathbf{p_2}]))$. If $[\mathbf{p}]$ is not ambiguous, this policy will tend to avoid classifying as indetermined. Experiments tend to indicate that this can improve the efficiency of the solver spectacularily when the anisotropy is severe.

4 Fuzzy systems as universal approximators

A fuzzy system is a set of *if-then* fuzzy rules that maps inputs to outputs [13]. Numerous approaches have been suggested to provide the best approximation that maps the fuzzy system inputs to its outputs. From the early 90s, several authors have established that different types of fuzzy systems possessed the "universal approximation" property [14,13,2]. These results claim that different fuzzy reasonning methods are capable to approximate arbitrary continuous function on a compact domain with any specified accuracy (see for further details [15,16]): most of these methods extract fuzzy rules from numerical data assuming the divisions of input variables into fixed regions [17].
Universal approximation results are *contradictory*. On one hand, approximating models exhibit exponential complexity in terms of number of components, *i.e.* the number of components grows exponentially as the approximation error tends to zero. On the other hand, if the number of components is bounded, the resulting set of models is *nowhere dense* in the space of approximated functions. As a consequence, we cannot approximate in general any continuous function arbitrarily well, if the number of components are restricted. Thus, only functions satisfying certain conditions can be approximated by such models, or, alternatively, we need unbounded number of components.
The following example illustrates these mutually contradicting results.

EXAMPLE **1 (**Kosko's fuzzy "universal" approximator) *Kosko showed in [18] that any real continuous function can be approximated by an additive fuzzy systems. The basic idea is to cover the graph of the function by rectangular fuzzy patches (see [13]). These fuzzy patches are equivalent to fuzzy* if-then *rules: less certain rules are large patches, more precise are small patches. The approximation can be achieved at any degree of accuracy by using a finite number of fuzzy patches [19]. Major disadvantages of such local approximators lie in:*

(*i*) partitioning the data*: partitioning methods definitely assigned each sample to one cluster. This method has the same disadvantages as k-*means *of being restricted to continuous data and of being a random algorithm subject to local minima [20].*
(*ii*) the shape of fuzzy sets*: what is the best shape for the MF? Fuzzy sets can have any shape, each shape affects how well a fuzzy system of* if-then *rules approximate a function.*
(*iii*) selecting explanatory variables*, which constitutes a means of choosing sets of*

optimally discriminating variables but decreases the quantity of information gathered by the fuzzy system. □

Kosko's approximation task, as described above, is then a *trade-off* between the specified accuracy and the number of components. This naturally raise the question to what extent the approximation should be *accurate*.
From the practical point of view, it may seem enough to have an "acceptably" good approximation, where the given problem determines the factor of acceptability in terms of accuracy.
Due to ambiguity and complexity of the identifying system, a range of techniques [21,22] attempt to take advantage of a thorough understanding of the system's behaviour and black-box approach, such that the known parts of the system are modeled using physical knowledge, and the less certain parts are approximated using process data with suitable approximation properties. For instance, Wang's fuzzy system [23] have a feedforward architecture that ressembles the feedforward multilayer neural systems used to approximate functions, by minimizing the difference between the fuzzy target vector and the actual fuzzy output vector. See also [13,24]. Such local optimization has several drawbacks:

1. No guarantee of convergence to the global optimum of the criterion can be provided,
2. If several values of the estimated parameters correspond to the same value of the criterion (such a situation may result from the fact that the parameters are not globally identifiable), the algorithm picks one of them without indicating that there are others,
3. The choice of a initial value for the parameters relies largely on guesswork,
4. In fuzzy problems, one is not actually interested in the optimal value of the parameters in the sense of a criterion but would rather like to characterize the set of all values that are acceptable in a sense to be specified.

A possible way out of problems 1 to 3 is to use deterministic global optimization methods such as those described in Belforte *et al.* [25], but this still leaves difficulty 4.
This is why we suggest an alternative approach - based on *interval analysis* - which can also be used to obtain guaranteed estimates of suitable characteristics of $\mathbb{S}$ in the nonlinear case.

5 Numerical experiments

We suppose in the following, without loss of generality, that MF are *triangle-shaped*. To illustrate the behaviour of our algorithm BISECT, we consider a two parameter estimation problem, which makes it posible to draw pictures of the paving obtained.

EXAMPLE **2 (**Fuzzy filter) *In this example, a simplified version of the problem explored by Jaulin and Walter* [1] *[6] is given. The vector comprising all available data* **y** *is:*

$$\mathbf{y} = (1.59, 1.44, 1.30, 1.18, 1.07, 0.96, 0.87, 0.79, , 0.71, 0.64)^T.$$

The numerical values of the corresponding interval data are given in Tab. 1.

[1] The extension of the method to multiple-output problems is straightforward.

t_i	y_i	y_i^-	y_i^+
1	1.59	1.18	2.00
2	1.44	0.62	2.26
3	1.30	0.40	2.21
4	1.18	1.09	1.27
5	1.07	0.32	1.81
6	0.96	0.12	1.81
7	0.87	0.71	1.03
8	0.79	-0.17	1.76
9	0.71	0.31	1.12
10	0.64	0.46	0.83

Table 2
Interval data.

The fuzzy set $\widehat{\mathbb{P}}$ *to be characterized consists of the fuzzy variables vector* $([p_1],[p_2])^T$ *such that the graph of the function :*

$$f(\mathbf{p},t) = p_2 \exp(-p_1 t), \tag{17}$$

crosses all the data bars of Figure 4. In this simulated example, the $[y_i]$ *have been computed by adding a random error interval with radius* $\rho_i = 0.5|y_i| + 1$ *to the* y_i*. The intervals* $[y_i]$ *may be seen as the zero-cut interval of the ith MF* μ_{Y_i}*, centered on* y_i *and supposedly triangle-shaped.*
The initial box domains for the parameters p_1 *and* p_2 *may be arbitrarily large, by example*

$$[p_1] = [-10000, 10000] \text{ and } [p_2] = [-10000, 10000], \tag{18}$$

i.e. no prior *information is available on the parameters. The feasible fuzzy set for the parameters is given by (16), where the search domain* $[\mathbf{p}](0)$ *is taken as* $[-1,5]\times[-5,5]$*. The coordinate functions of* f *are given, for each* α*−cut, by:*

$$y_i^\alpha = p_2^\alpha \exp(-p_1^\alpha t), i = 1,\ldots,10. \tag{19}$$

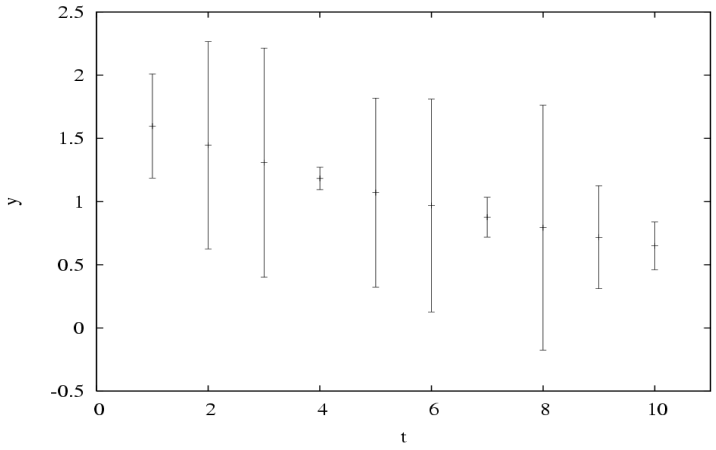
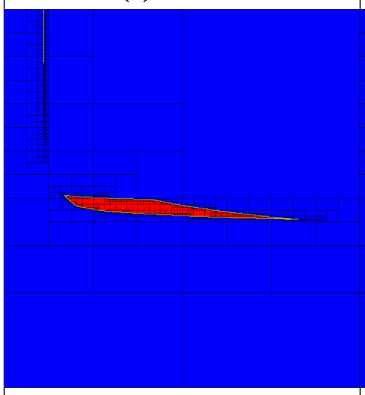
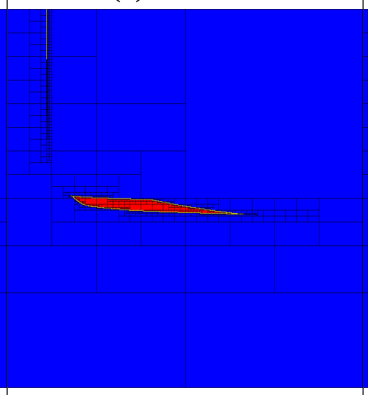
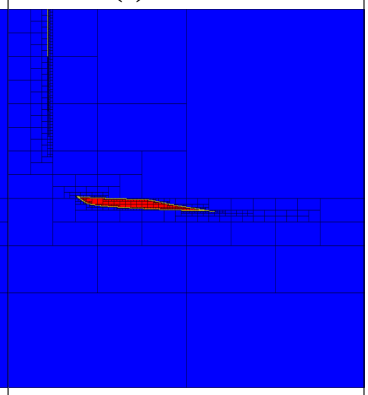
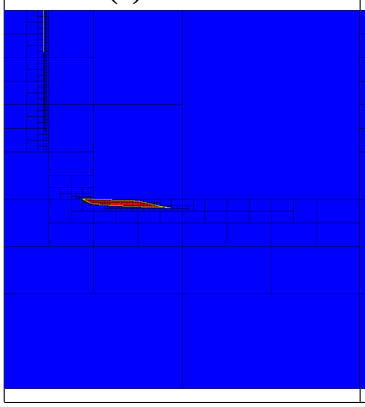
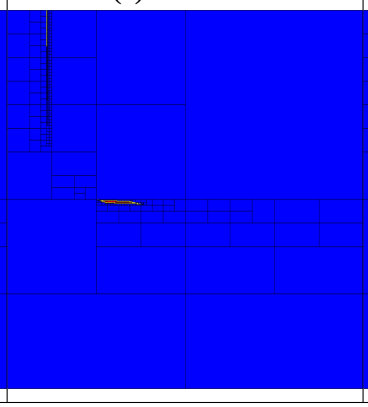
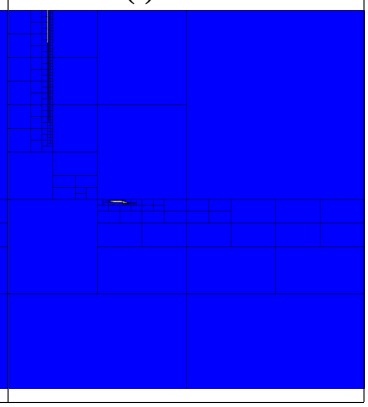


Figure 4. Measurement times and corresponding interval data.

In less than 3s, on a PENTIUM *IV,* BISECT *generates the pavings of Tab. 3, thus bracketing the posterior feasible set for* $\mathbf{p}_\alpha$ *between inner and outer approxima-*

tions. Figures 3.a-f gives top-view representation of the MF of $\mathbf{p}$ *for some values of* α, i.e. *the joint MF at the level* α *of* p_1 *and* p_2 *that is consistent with the equation (19) and the domains (18). A 3-dimensional plot would show the fuzzy domain as a barn like structure rising up from its base.*

Table 3
Top-view of the paving generated by BISECT to bracket the solution set (in red) for each α-value in the parameter space. The outer frame corresponds to the box $[-1,5]\times[-5,5]$.

(a) $\alpha = 0$	(b) $\alpha = 0.1$	(c) $\alpha = 0.2$
		
(d) $\alpha = 0.4$	(e) $\alpha = 0.7$	(f) $\alpha = 1$
		

EXAMPLE **3 (**Fuzzy function approximation of a MISO^2 model) *The curvature* $\kappa(t)$ *of an arbitrary twisted curve* $\mathcal{C}$ *measures the rate of change of the tangent when moving along the curve. It measures, so to speak, the deviation of the curve from a straight line in the neighbourhood of any of its points. It is quite easy to derive an analytic expression of the curvature which is valid when* $\mathcal{C}$ *is represented by an allowable parametric representation* $\mathbf{x}(t)$*:*

$$\kappa(t) = \frac{\mathbf{x}' \times \mathbf{x}''}{|\mathbf{x}'|^3}, \tag{20}$$

where $\times$ *denotes the vector product. Derivatives with respect to time are denoted by primes,* e.g. $\mathbf{x}' = \frac{d\mathbf{x}}{dt}$ *and* $\mathbf{x}'' = \frac{d^2\mathbf{x}}{dt^2}$*. (20) is equivalent to (see [26] and references therein):*

$$\kappa(t) = \frac{\sqrt{(\mathbf{x}' \cdot \mathbf{x}')(\mathbf{x}'' \cdot \mathbf{x}'') - (\mathbf{x}' \cdot \mathbf{x}'')^2}}{(\mathbf{x}' \cdot \mathbf{x}')^{\frac{3}{2}}}, \tag{21}$$

When $\mathcal{C}$ *is a curve of radius R (see figure 3.a) has the following cartesian represen-*

tation [3] *:*

$$\mathbf{x}(t) = (R\cos nt\cos t, R\cos nt\sin t, R\sin nt) \tag{22}$$

with t as parameter, it is easy to obtain from (21) a simpler representation of the curvature:

$$\kappa(t) = \frac{\sqrt{-n^4\cos^2(nt) + n^6 + 4n^2\cos^2(nt) + 4n^4 + cos^4(nt) - n^2cos^4(nt)}}{R\sqrt[3]{n^2 + \cos^2(nt)}}. \tag{23}$$

Figure 3.b plots the domain of $\kappa(t)$.

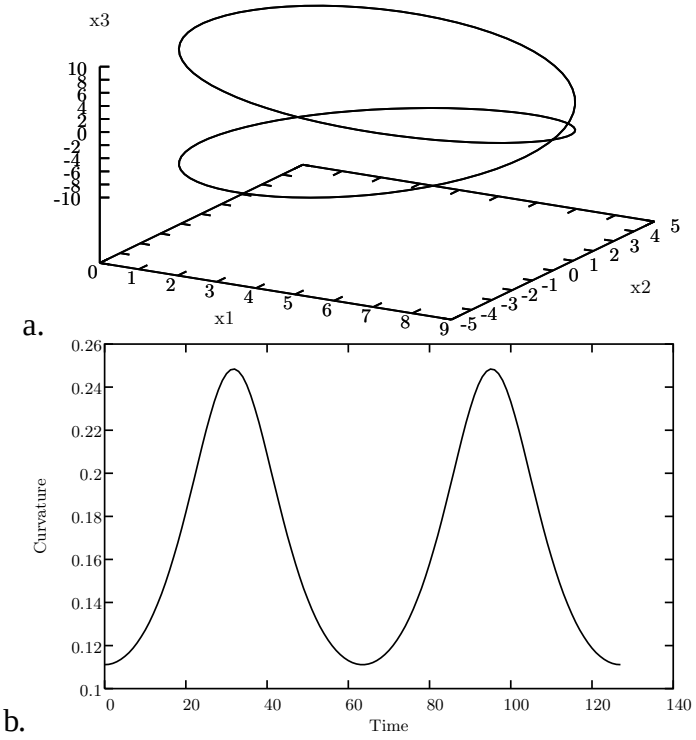


Figure 5. a. 3-dimensional representation of the Clelia curve. b. Graph of the function $\kappa(t)$ given by equation (23).

A first approach may consider the values of the independent variables as uncertain and modelizes the uncertainty with fuzzy variables. Assume that a set of N input-output data pairs $(\mathbf{x}_i, \kappa_i)_{i=1}^N$ *is available. Recall that* $\mathbf{x}_i \in \mathbb{R}^3$ *are vectors and* κ_i *is scalar. The data set is split into a training and a checking data sets. The fuzzy inference system (FIS) used is a Sugeno-type, implemented in the C programs of* `ANFIS` *architecture proposed in [27,28].* `ANFIS` *uses a hybrid learning algorithm to identify the parameters of Sugeno-type FIS. It applies a combination of the least-squares method and the backpropagation gradient descent method for training FIS memebership function parameters to emulate a given data set.*
A graphical representation of the corresponding MFs is given Fig. 3. Notice that the MFs associated to each regressor form a fuzzy partition of the 3d-space.

The simulation detailled in Fig. 3 tests the system identification accuracy by comparing the curvature value κ *with the output of the fuzzy system. Fig. 3.a depicts the training error performed on the training data set. With increased number of MFs,*

[3] *Clelia* curve was studied by Guido Grandi in 1728.

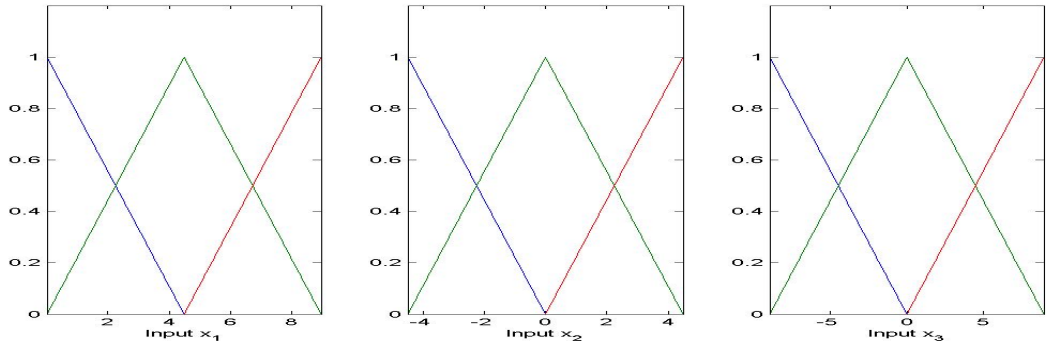


Figure 6. Membership functions of independent variables in **x**.

the resulting fuzzy system contains narrower input MFs and accuracy of the resulting fuzzy system with respect to the training data set is improved, but the ability of the fuzzy system to generalize for inputs (test set) may be degraded as illustrated in our results (Fig. 3.b). In a similar way, descreasing the output error improves the accouracy with respect to the training data set, but accuracy in the presence of inputs different than the training data set is degraded. These tests illustrate some effects in parameter choice on the resulting fuzzy system.

A second approach consists to use interval analysis tools. In this simulated example, we assume that the values taken by κ *are uncertain and sampled at the rate* $n\frac{\pi}{11}$*: the prior intervals* $[\kappa_i]$ *are computed by adding a centered error interval to the associated measurement* κ_i*. The set* $\widehat{\mathbb{S}}$ *to be characterized consists to all the values of* $\mathbf{p} = (R,n)^T$*. such that the graph of the function* $\kappa(t)$ *crosses all the data bars of figure 3. The dataset is made of 22 data.*

For $\epsilon = 0.03$, BISECT *generates the subpaving represented on Figures 4.a-f. in 7s on a* PENTIUM IV*, consistent with the equation (23).. The prior box for the parameters is taken as* $[\mathbf{p}] = [-10,10]\times[-10,10]$.

More interesting, one can know generate the posterior feasible set for the κ_i *(posterior estimates) from the "simulator" (23).*

5.1 Discussion

The success of any identification method relies on the descriptive power of the model structure as well as on the quality of the estimation data. Compared to a pure data driven identification approach, it is easier to avoid data caused pitfalls by using an expert determined model structure.
Concerning estimation algorithms, it is worth stressing that the scheme of section 3.2 is robust in the sense that the fit of the tuned model is at least as good as what is obtained with classical approach. A distinct advantage with interval analysis is that redundancy in terms of similar MFS can be avoided since the parameter in explored exhaustively. Apart from reducing the model complexity with respect to FIS, this also leads to less ill-conditioned problems. Realistic advantages can be found compared to the statistical approach:

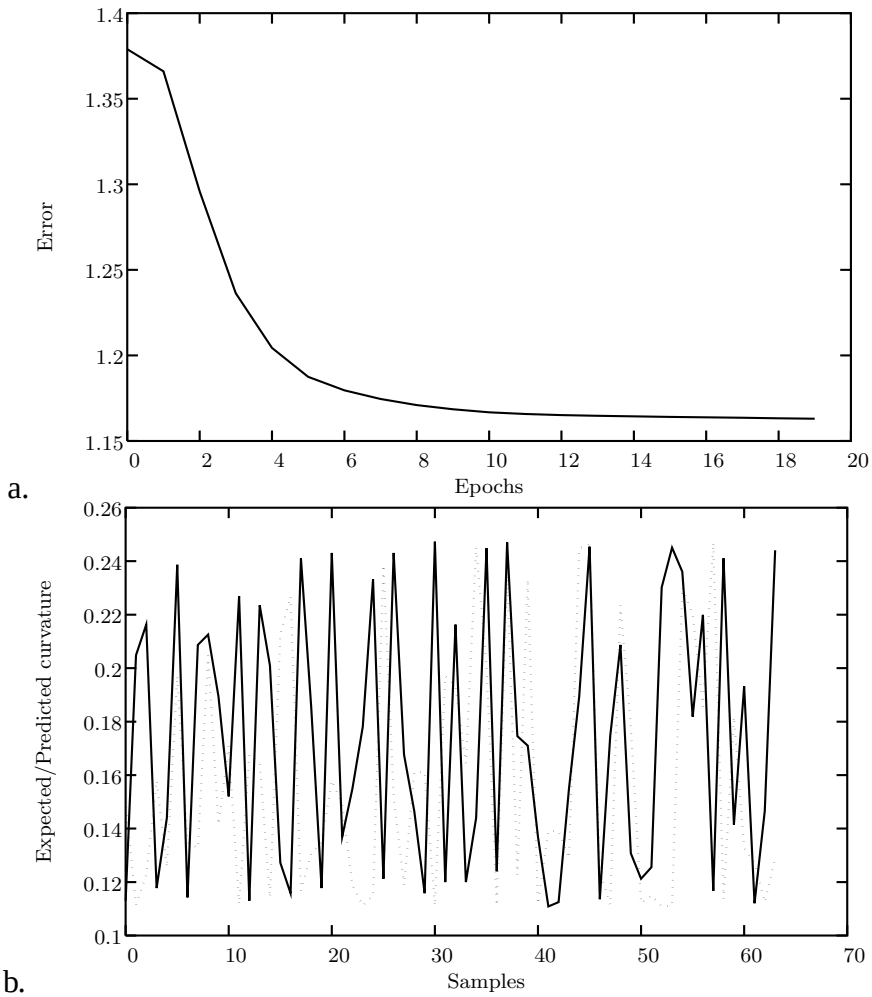


Figure 7. a. Measured outputs (–) and simulated outputs (. . .) computed by the Sugeno-type FIS program `ANFIS`.

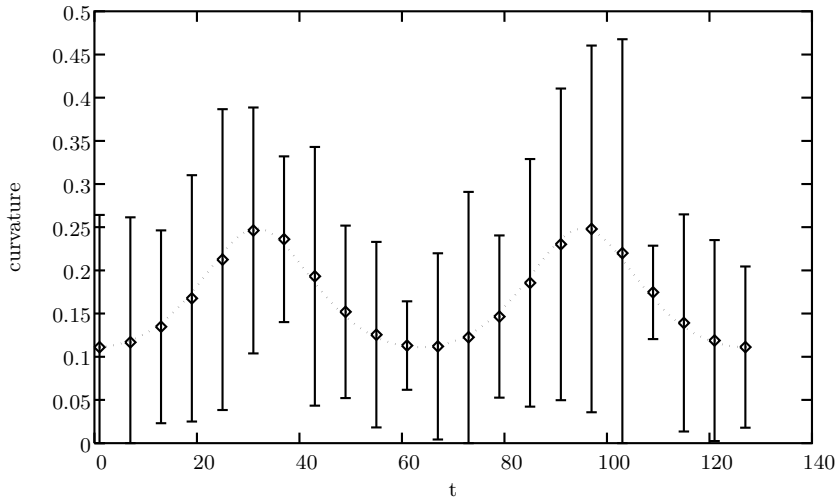
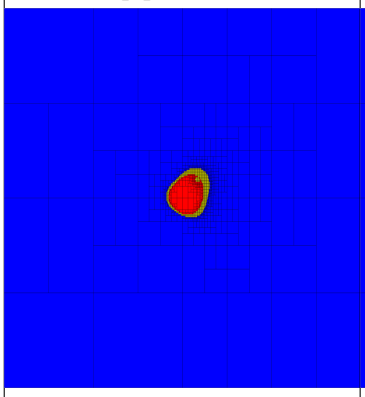
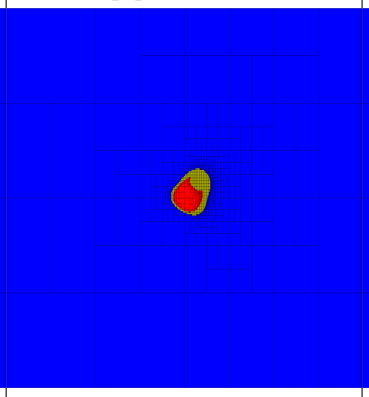
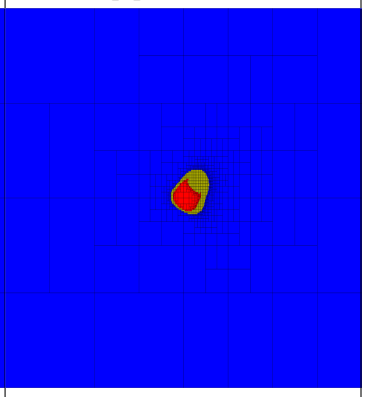
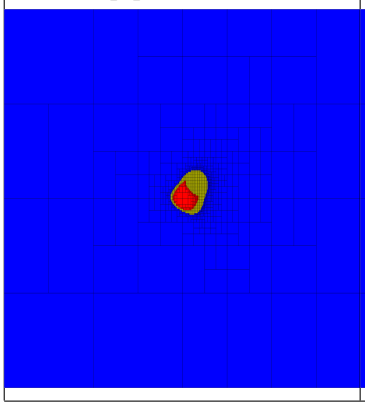
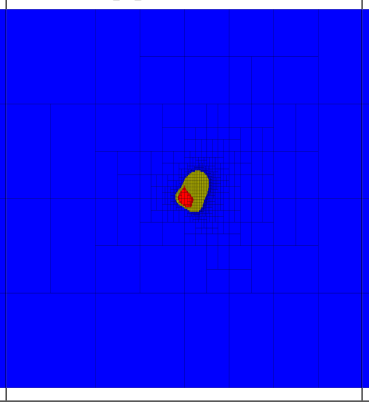
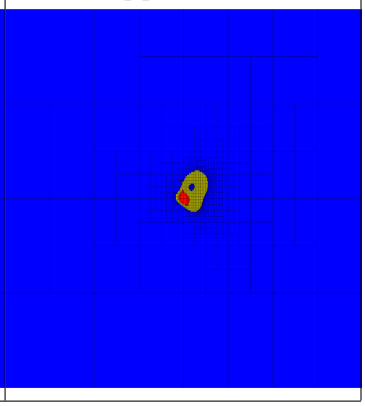


Figure 8. Experimental data (♦) together with their uncertainty intervals and graph of the function $\kappa(t)$ (see Eq. (23)).

Table 4
Top-view of the paving generated by the algorithm to bracket the solution set (in red) for each α-value. The outer frame corresponds to the box $[-10, 10] \times [-5, 5]$.

α-cut value		
[a] $\alpha = 0$	[b] $\alpha = 0.2$	[c] $\alpha = 0.4$
		
[d] $\alpha = 0.6$	[e] $\alpha = 0.8$	[f] $\alpha = 1$
		

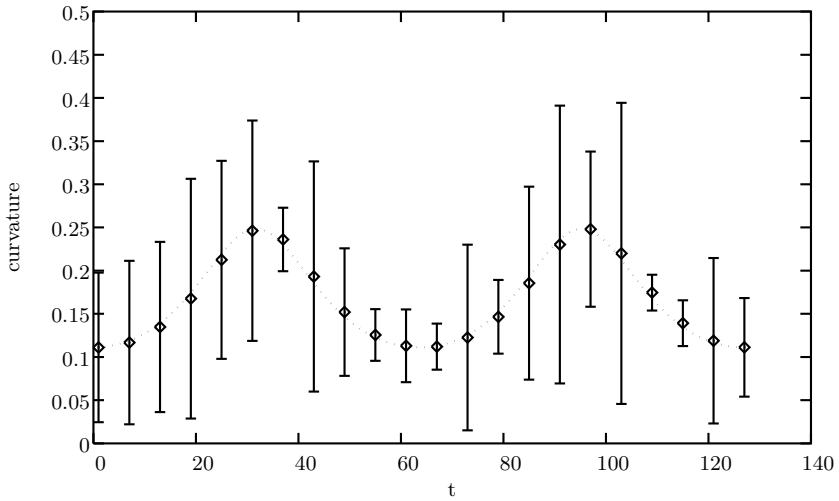


Figure 9. Posterior feasible intervals for the κ_i superposed on the graph of the function $\kappa(t)$.

(1) The error structure is quite simple and similar information usually available in most practical cases, not assuming *any a priori* statistical information about the error.
(2) The computation of the parameter domain is conceptually simple and is practically feasible even if the number of data is not large.
(3) The algorithm is *deterministic*.

Least-Square estimation suffers from the fact that the cost function to be minimized is a sum of terms involving the same parameters, so multioccurence of these parameters is unavoidable and tends to make inclusion functions for the cost function very pessimistic, which complicates the elimination of interesting parts of the search domain. A distinct advantage is that redundant parameters can be avoided by decomposing a fonction in several primitives involving a single operator. For instance, the constraint $x_1 \exp(x_2) + \sin x_1$ can be decomposed into the following set of primitive contraints

$$\begin{cases} a_1 & = \exp(x_2) \\ a_2 & = x_1 a_1 \\ a_3 & = \sin x_1 \\ a_4 + a_3 & = 0 \end{cases} \tag{24}$$

The domains associated with all intermediate variables (here a_1, a_2 and a_3) are $]-\infty, +\infty[$. A method for estimating x_1 and x_2 with respect to $x_1 \exp(x_2) + \sin x_1$ is to contract each of the primitive contraints in (24) until the contractors become inefficient.

6 Conclusion

In this paper, we proposed an approach to the fuzzy approximation problem based on interval analysis.
The problem of estimating the parameter fuzzy sets of a (non)linear model from prior knowledge, experimental data and collateral requirements is viewed as one of set inversion, which is solved in an approximate but guaranteed way with the tools of interval analysis. It is possible to charaterize the fuzzy set of all parameter vectors that are *consistent* with the data in the sense that the errors between the data and corresponding model outputs fall within known prior bounds. This has been illustrated on simple simulated examples for time-invariant models whose outputs are linear in their inputs, even if nonlinear in their parameters.
No monotonicity or convexity assumptions needs to be made on the concerned membership functions. Upon completion of the algorithm, a paving bracketing the contours of the solution membership functions is found (or not) with a precision controlled by the solver.
We don't formulate the available knowledge in terms of fuzzy if-then rules. No test set nor validation step (as in classical machine learning theory) is needed: the model does meet the expected performance when the model is suited to the problem, even in the case of a small data set.
The computation process has drawbacks: (*i*) its complexity is exponential in the number of parameters which restricts its use to low-dimensional problems, (*ii*) the algorithm presented here is far from optimal from the viewpoint of computational time and significant improvements can be expected in the near future, (*iii*) efficient functions are needed which are available only when an explicit solution for the equations defining the model can be found.
The advantages are numerous: first, the solver characteristics are different from,

for instance, the Kosko statistical approach that requires a (large) set of data points through which the constructed patches pass. Here, prior knowledge such as prior ranges is sufficient and collateral requirements such as the shape of the membership function have to be provided to the fuzzy system. Second, It is now possible to obtain guaranteed estimates of fuzzy parameters even when these parameters are not identifiable. Nonlinear constraints are easily handled. This approach could be extended to the case where X and Y consists of infinitely many variables, whatever could be the type of membership function to be used.

References

[1] Y.H.O. Chang and B.M. Ayyub, "Fuzzy regression methods - a comparative assesment," *Fuzzy Sets and Systems*, vol. 119, pp. 187–203, 2001.

[2] L.X. Wang, "Fuzzy systems as universal approximator," in *IEEE Int. Conf. on Fuzzy Systems*, San Diego, 1992, pp. 1163–1169.

[3] V. Vigneron, H. Maaref, C. Barret, and K.. Kallel, "Guaranteed fuzzy function approximation using interval analysis," in *IEEE International Conference on Systems, Man and Cybernetics*, Hammamet (Tunisia), October, 6-9 2002.

[4] D. Dubois and H. Prade, "Fuzzy intervals versus fuzzy numbers: is there a missing concept in fuzzy set theory," in *Proc. of the 25th Linz seminar on Fuzzy Set Theory*, Linz, Austria, January 2005.

[5] D. Dubois and H. Prade, "Random sets and fuzzy interval analysis," *Fuzzy Sets and Systems*, vol. 42, no. 1, pp. 87–101, 1991.

[6] L. Jaulin and E. Walter, "Guaranteed nonlinear parameter estimation from bound-error data via interval analysis," *Mathematics and Computers in Simulation*, vol. 35, no. 2, pp. 123–137, 1993.

[7] E. Walter, Ed., *Special issue on parameter identification with error bounds*, vol. 32, 1990.

[8] B. Kosko, *Fuzzy thinking, the new scince of fuzzy logic*, Hyperion Books, 1994.

[9] D. Dubois, E. Kerre, R. Mesiar, and H. Prade, *Fundamentals of fuzzy sets*, chapter Fuzzy interval analysis, pp. 483–581, Kluwer, 2000.

[10] R.E. Moore, *Interval analysis*, Prentice Hall, New York, 1966.

[11] L. Jaulin, *Solution globale et garantie de problèmes ensemblistes. Application à l'estimation non-linéaire et à la commande robuste.*, Phd thesis, Université Paris-Sud, Orsay, France, 1994.

[12] J.P. Norton, "Special issue on bounded-error estimation: Issue 1.," in *International Symposium Control and Signal Processing*, 1990, pp. 1–118.

[13] B. Kosko, "Fuzzy systems as universal approximators," *IEEE Trans. on Computers*, vol. 43, no. 11, pp. 1329–1335, November 1994.

[14] J.L. Castro, "Fuzzy logic controllers are universal approximators," *IEEE Trans. on SMC*, vol. 25, pp. 629–635, 1995.

[15] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modelling and control," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[16] M. Sugeno, "Structure identification of fuzzy models," *Fuzzy Sets and Systems*, vol. 28, pp. 15–33, 1988.

[17] R.R. Yager and D.P. Filev, "Unified structure and parameter identification," *IEEE Trans. on Syst. Man and Cyb.*, vol. 23, no. 4, pp. 1198–1205, 1993.

[18] B. Kosko, "Fuzzy fonction approximation," in *IEEE International Joint Conference on Neural Networks*, Baltimore, 1992, vol. 1, pp. 1209–1213.

[19] J.A. Dickerson and B. Kosko, "Fuzzy function approximation with ellipsoidal rules," *IEEE Trans. on Syst. Man and Cyb.*, vol. 26, no. 4, pp. 542–560, 1996.

[20] J.C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, pp. 32–57, 1974.

[21] H. Ishihashi, "Iterative fuzzy modeling and a hierarchical network," in *Proc. 4th IFSA Congress*, Brussels, 1991, pp. 49–52.

[22] H. Nomura, I. Hayashi, and N. Wakami, "A self-tuning method of fuzzy control by descent method, a hierarchical network," in *Proc. 4th IFSA Congr.*, Brussels, 1991, pp. 155–158.

[23] L.X. Wang and J.M. Mendel, "Back-propagation fuzzy system as nonlinear dynamic system identifiers," in *Proc. IEEE Int. Conf. on Fuzzy System*, San Diego, 1992, pp. 1409–1416.

[24] P.V. Krishnamraju, J.J. Buckley, K.D. Reilly, and Y. Hayashi, "Genetic learning algorithm for fuzzy neural nets," in *Proceedings of 1994 IEEE International Conference on Fuzzy Systems*, Brussels, 1994, pp. 1969–1974.

[25] G. Belforte, B. Bona, and V. Cerone, "Parameter estimation algorithms for a set-membership description of uncertainty," *Automatica*, vol. 26, no. 5, pp. 887–898, 1990.

[26] E. Kreyszig, *Differential geometry*, Dover, 1991.

[27] J.-S. Jang, "Fuzzy modeling using generalized neural networks and kalman filter algorithms," in *Proc. of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, 1991, pp. 762–767.

[28] J.-S. Jang, "Anfis: Adaptive-network-based fuzzy inference systems," *IEEE Trans. on Systems, Man and Cybernetics*, May 1993.